DCC++ EX Project Meeting Notes

Monday, June 1 2020 - 11am to 1pm EST

Attendance

Larry Dribin
Mike Dunston (Atani)
Roger Beschizza
David Cutting
Fred Decker (FlightRisk)
Harald Barth (Haba)
Anthony Williams (Dex)
Kevin C. Smith (KC Smith)
Gregor Baues (grbba)

Resources

Developers

- Mike Dunston
- David Cutting
- Harald Barth
- Chris (UKbloke)
- Fred Decker

Managers (Project Organizing, Documentation)

- Fred Decker (Project Lead)
- Roger Beschizza
- Kevin C. Smith

Installer Project

- Anthony Williams (Dex)

JMRI Integration

- Harald Barth

- David Cutting
- Anthony Williams
- Fred Decker

Topics

1. **Deploy the waveform/packet code as a library or included unit?**

The key considerations are simplicity, ease of use and solid, reliable operation. A library would require another piece to install and hide the code behind one more layer from tinkerers, but keep the code a base from which all versions of "DCC++" products could derive. Having the new DCC++ EX Installer somewhat mitigates concerns of complexity, since everything can be taken care of automatically.

Action: *Agreed that we would pursue the library approach and use the unit approach as a fallback if we run into any unforeseen difficulty. Assigned - Chris, David, Harald*

2. **Coding Standards**

Have a need to standardize code layout, comments, etc.

Action: *Will agree on standards and put them in a "CONTIBUTING.md" file on Github. Example: http://mozillascience.github.io/working-open-workshop/contributing/ Assigned - all developers*

Action*: In particular, the overly verbose comments in the code were seen as hurting code readability. Those comments will be moved to our technical document and operations manual. Assigned - Fred, Roger*

3. **Solving issue of huge threads/difficulty finding information**

Action*: Will have the main links and "how to get started" pinned as a sticky in Trainboard. Assigned - Fred, Roger*

Action*: Will use private message, Discord, GitHub and conference calls to handle some of the more technical developer issues to keep Trainboard for more general and wider interest issues.*

4. **Board Compatibility / Hardware Platform**

Action: *Will try to maintain Arduino Uno compatibility where possible, but the Mega will be the recommended platform. Where the Uno either lacks the capability to handle a specific function,* *or a choice must be made based on available memory (an either/or situation between features) we will make that clear in the notes and the installer. Assigned - All developers*

5. **GPIO Pins used/numbering**

Compatibility is important for users already familiar with DCC++ Classic, however there may be a need for breaking changes for things like pin use. There is already confusion in that the original DCC++ itself changed pin jumper positions. Our new “1 timer” approach means that we can use any pin for DCC waveform generation. The downside of this is that it has to be clearly documented. The advantage is that we would not need jumpers!

Action: *We will try the jumperless approach and just change pins in the configuration file as needed. Assigned - David, Chris, (Mike Advisory)*

6. **The One Timer, One Interrupt approach**

The new DCC++ EX packet generation API being developed primarily by Chris and David is using 1 timer with one interrupt and doing in code what we were doing with hardware timers. The advantage of the timers is precise timings of the DCC waveform 1s and 0s. But the advantage of using code is simplicity and freeing up a timer for something else like the Railcom cutout. We are using a fast write library to handle the pin on/off, pulse generaton, but because of how this works, if we set the single timer for 58us for a “1”, then a “zero” will need to be 116us instead of 100. Potential issues?

Action: We will test and keep an eye out for any issues with the new timer and packet generation API. This is within the tolerance for timing, but we need to check it against as many decoders as possible. Other things to test will be logic analyzers and DCC packet decoders to see if they still read the packets correctly. Assigned - Everyone with something they can test.

7. **Confusion of how to power the Base Station**.

*Action: In all documents and images, make clear the difference between DC, AC and the DCC bipolar waveform. In particular, we need to show that the input power to both the Arduino and motor shield is **DC**! Assigned - Fred, Kevin, Roger*

8. ***Loconet/LCC/CMRI/S88 support***

Action: The DCC++ EX project will prioritize and include a railcom cutout and look at implementing actually reading the Railcom data as an added function of the Base Station. Other protocols to be revisited in the future. Assigned - Chris, David, Mike

9. **Networking Support**

Action: *We will retool the network code for the Arduino WiFi shield, ESP8266, etc. to be its own abstraction layer. In order to simplify things, we may look to use the code that comes with the 8266 and have our bridging layer communicate with DCC++ EX with AT Commands (or similar) Assigned - Fred, David, Mike*

10. Controller

We need a controller to complete the BaseStation. Gregg's original software that requires the Processing IDE is not viable. We need a full function throttle. Is touchscreen a possibility?

Action: Continue discussion on creating an Arduino Shield and/or a WiFi controller solution similar to the Digitrax Zephyr and Dave Bodnar's handheld controller. Assigned: David, Fred

11. Method for handling turnouts, outputs, etc. (Related to 10 above)

JMRI can handle direct control over available GPIO pins, but there is no "network bus" to handle external devices using serial lines. Do we use a Railcom based solution, some other existing bus, I2C, RS485, something else? I2C and a different backbone?

Action: *Explore options. Create a test case for I2C and the PCA9685 16 Channel Servo Driver. Differential I2C has a longer range so could work. Assigned - David, Fred*

12. JMRI Integration

We will have to work on the JMRI code to handle changes we are making. Particularly important is the Auto-Reconnect feature for making sure everything, especially the USB/Network connects reconnects when the Base Station resets for any reason. Some boards reset only the CPU, leaving the serial connection still running, others reset the com port too. JMRI will report this as a yellow color in the power manager.

Gregor mentioned wanting to send a URL with a command. An IOC Abstraction layer with parameters. David talked about using a websocket like Atanis' ESP32. Gregor countered with a suggestion about using REST.

Action*: Look at the JMRI source to see what method is implemented now regarding reconnect. This may be just a case of reducing the time between reconnect attempts. Check the power manager and current reading code. Assigned - David, Fred, Harald*

Action: *Find TwinDad. His input would be invaluable. Assigned - Fred.*

Action*: Discuss further Gregor's ideas on how to send commands.*

13. Modbus support

Could we use Modbus as a Comm channel for the user interface command input and response?

Action*: There are mixed feelings about this as a solution, but will give it further study. Assigned - Everyone*

14. Software Switching of Main and Program Tracks

People have used a DPDT switch to handle this mechanically when needed. Is there a need to be able to switch this programmatically?

Action*: Agreed there is a need. Will add the feature and test. Assigned - Chris, David*

15. Current Sense and ACK Detection

We have progressed far on this. Between Mike and Fred's work, Harald's major contribution and now Chris and David reworking the DCC "Engine", we can focus on getting this right.

Action: Finish the code to account for multiple voltages, current sense devices, and ADC resolutions and test. Assigned - Fred, David, Chris. Test: Everyone.

16. Accessory Decoder Addressing

This was not covered in the meeting, but will be discussed next meeting. Fred worked on this and fixed issues in the DCC protocol decoder for Sigrok Pulseview. We currently use the 2 part (address/subaddress) method instead of linear addressing. Address 229 would be 58, 0. Then 1 throws the "switch" and 0 "closes" it. We can either add to the existing commands to handle both methods of addressing, or add a new letter for a new command to handle linear addressing. The first option may require a JMRI change.

17. Installer Update

Dex reported that he is now using the Arduino CLI which considerably simplifies his build process and the user experience. As always, further testing is appreciated.

18. Evangelizing

Action: *We will enlist people like Geoff Bunza, Dave Bodnar and others for their input and to help promote DCC++ EX when it is the kind of product that would be worthy of their (and our) recommendation. Assigned - Fred*

19. New Slogan!

“We’ll let you tinker with our DCCpp!”

Say that quickly out loud! Ok, it needs work ;) But a major advantage of DCC++ EX is the ability to change and modify functionality in addition to settings. You can fix things almost immediately without waiting for a manufacturer to issue an update. You can’t do that with a commercial system.